AMENDMENTS TO THE SPECIFICATION

On page 1, please replace the subheading "FIELD OF THE INVENTION" with "FIELD".

On page 1, please replace the subheading "BACKGROUND AND SUMMARY OF THE INVENTION" with "BACKGROUND AND SUMMARY".

On page 5, please replace the subheading "DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS" with "DETAILED DESCRIPTION OF THE VARIOUS EMBODIMENTS".

Applicant notes the original application was filed without paragraph numbers. To facilitate the incorporation of these amendments, Applicant has numbered the paragraphs. Please replace the following paragraphs rewritten in amendment format:

[0001] The present ~~invention~~disclosure relates to improved software simulation systems.

[0006] Through the RTI interface, the independent but interoperating simulations (the virtual environment simulation and the functional simulation) communicate with each other via objects that are published and subscribed there between. Preferably, the RTI interface includes an application program interface (API) plug-in module for this purpose. Thus, the present ~~invention~~disclosure provides the ability to communicate with or affect a virtual environment from an external functional simulation through a standard interface.

[0007] According to one aspect of the present ~~invention~~disclosure, disclosed herein is a method of simulating the behavior of a user-interactive environment, the

method comprising: (a) running a virtual environment (VE) simulation application that (1) graphically depicts a VE, (2) receives input from a user that corresponds to a user interaction with the VE, and (3) provides graphical output to the user that corresponds to a condition of the VE; (b) running a functional simulation application that determines the condition for the VE at least in part based upon the user input; (c) communicating the user input received by the VE simulation application to the functional simulation application via a high level architecture (HLA) protocol; and (d) communicating the condition determined by the functional simulation application to the VE simulation application via the HLA protocol.

[0008] According to another aspect of the present ~~invention~~disclosure, also disclosed herein is an apparatus for interfacing software configured to graphically depict a three-dimensional user-interactive virtual environment (VE) with software configured to functionally simulate an application associated with the VE, the apparatus comprising: (a) a library of HLA objects that correspond to a state of the VE that is dependent at least in part on user interaction with the VE; and (b) a processor configured to (1) publish via RTI messaging at least one HLA object to the functional simulation software according to the HLA protocol, (2) subscribe via RTI messaging to at least one HLA object published by the functional simulation software according to the HLA protocol, wherein the subscribed HLA object defines at least in part a subsequent state for the VE, and (3) provide data derived from the subscribed HLA object to the VE software for processing thereby.

[0009] The present ~~invention~~disclosure can be implemented through computer executable instructions on any suitable computer-readable medium including

but not limited to a compact disc (CD), a computer hard drive, network-accessible server, or any other known storage device capable of storing executable programming code.

[0010] These and other features and advantages of the present ~~invention~~disclosure will be in part pointed out and in part apparent upon review of the following description, figures, and claims.

[0011] Figure 1 is a block diagram overview of a preferred embodiment of the present ~~invention~~disclosure;

[0014] Figure 1 is a block diagram overview of an ~~preferred~~ embodiment of the present ~~invention~~disclosure. The simulation system 100 comprises a user-interactive virtual environment (VE) simulation application 102, a functional simulation application 104, and a runtime infrastructure (RTI) interface 106 that manages communications between the VE simulation application 102 and the functional simulation application 104.

[0015] The VE simulation application 102 graphically depicts an environment with which a user of the system 100 can interact. Preferably, this environment is graphically depicted as a three-dimensional (3D) environment. Examples of preferred 3D environments for the present ~~invention~~disclosure include: aircraft and their maintenance environments, including crew station control panels, avionics, electrical connectors, avionics and other system components that visually or audibly convey status, and test equipment. Other 3D environments to which this ~~invention~~disclosure might apply would be any complex system or vehicle requiring simulation for

maintenance training, including reusable manned spacecraft, submarines, large water vessels, and the like. The VE simulation application is preferably implemented on a desktop personal computer (PC). Preferably, such a PC is properly equipped and configured for a network interface, standard user inputs, and high-end 3D graphics computation and display. Additionally, the ~~invention~~disclosure could be implemented on PC equipment properly configured to support immersive VR and related components, such as the associated head-mount display and cyber glove interface devices used therefor.

[0016] As the creation and design of 3D interactive virtual environments are well known in the art, further elaboration upon their inner details are not necessary in setting forth the present ~~invention~~disclosure. However, it should be noted that user interactions with the virtual environment (such as, with reference to Figure 1, flipping APU switch 108 to either the "on" position 110 or "off" position 112, or flipping the engine crank switch 114 to either the "L" position 116, "R" position 118, or "off" position 120) are triggers that signal a user input. This user input is used to determine any state changes that are necessary for the virtual environment to indicate back to the user (such as, for example, whether the "ready" light 122 should be illuminated).

[0017] The functional simulation application 104 operates to model the behavior of the system being emulated by simulator system 100. The operation of the functional simulation application 104 for a given end use of the present ~~invention~~disclosure is also known in the art, and as such, no particular functional simulation application is preferred by the inventors. Functional simulation applications are typically realized with real time software development technologies, which currently

use C, C++ programs to model functionality such as vehicle functionality in a flight simulation. These programs can be derived using routine skills in the art from existing operational flight program software, test or support equipment software, and the associated engineering design process.

[0019] The improvement ~~of~~ according to one of the ~~preferred~~various embodiments of the present ~~invention~~disclosure resides in part in the integration of these two independent but interoperating applications. According to the present ~~invention~~disclosure, communications between the two applications are managed by the RTI interface 106. RTI interface 106 communicates data between the applications 102 and 104 (referred to as federates, with the system 100 being a federation) via the High Level Architecture (HLA) protocol. An HLA interface 130 interfaces each application with the RTI messaging.

[0021] According to the present ~~invention~~disclosure, an application program interface (API) plug-in module 140 is created for the purpose of initiating and maintaining communications between applications 102 and 104 via the HLA-based RTI interface 106. The plug-in module 140, which is preferably implemented within the HLA interface 130 on the virtual environment (VE) side of the system 100 as shown in Figure 2, dynamically links methods supporting the HLA functionality that are developed using the virtual simulation API into the virtual simulation execution.

[0024] Generalizations necessary in the subscribed and published objects can be efficiently characterized using the Object Model Development Tool (OMDT) that is known in the art. The use of OMDT results in minimal software development to facilitate visual simulation authoring. For example, from a large set of data types available in OMDT, a small set of data types can be chosen by a practitioner of the ~~invention~~disclosure that is specifically suited to the general design requirements of the integration of the VE and functional simulations. Thus, two general aircraft cockpit simulation classes might be described using OMDT as simply "switch" and "indicator" classes with possible states being limited to ON and OFF for both. With OMDT, a naming or ID convention can also be chosen to satisfy overall system design requirements for uniquely identifying specific switch and indicator object instances within the system. With such an object model, source code builds and compiler code builds can be generated to produce a single re-usable plug-in that is generalized and valuable for use in many simulators.

[0025] This aspect of the ~~invention~~disclosure diminishes or eliminates for recompilation of the API plug-in module for various VE simulators, and further allows for the definition and identification of HLA objects during VE simulation authoring to be synchronized to a specific functional simulation's objects through simple text-naming discipline. The HLA classes defined through OMDT become the common data structure bridging any data type differences that may exist between the VE simulation and the functional simulation. Thus, the API plug-in module can be re-usable for different VE simulation applications if the library of HLA object supported by the module is sufficient to cover the different VE simulation applications. While it is possible that the API plug-in

module can also be re-usable for different functional simulation applications, it is expected that each functional simulation application will use its own associated API plug-in module within the VE simulation application, although this need not necessarily be the case.

[0031] A summary of software development requirements that characterize an ~~preferred~~ implementation of the API plug-in module are as follows:

- The software development process ~~should~~ provides functionality through a user interface that is conveniently presented with a 3D visual simulation application;
- The software development process ~~should~~ provides creation of one or more unique HLA objects with a state (e.g., "on" or "off", etc.) that is controlled by the user interface within the 3D visual simulation. The state of these objects ~~should be~~is published to the functional simulation application, and ~~should be~~is identifiable by the same unique name in each simulation;
- The software development process ~~should~~ provides the functionality of setting the states of all published objects upon events that are triggered within the 3D visual simulation;
- The software development process ~~should~~ provides creation of one or more unique HLA objects with states that are controlled by the functional simulation. The state of these objects are subscribed to and from the

functional simulation application, and must be identifiable by the same unique name in each simulation; and

- The software development process ~~should~~ provides for the virtual environment simulation application to catch the state changes of subscribed HLA objects, and thus trigger or influence changes of behavior within the graphically-depicted 3D virtual environment.

[0032] While the present ~~invention~~disclosure has been described above in relation to ~~its preferred~~various embodiments, various modifications may be made thereto that still fall within the ~~invention's~~disclosure's scope, as would be recognized by those of ordinary skill in the art upon review of the teachings herein.

[0035] As such, the full scope of the present ~~invention~~disclosure is to be defined solely by the appended claims and their legal equivalents.